

QUANTA

VOLUME 3 ISSUE 1

FEBRUARY 1986

CONTENTS

2. Information on the Group
3. Editorial
3. Members letters
6. Problem section
10. Archive backup
10. Book review: "Inside the Sinclair QL"
11. Supercharge review
14. Arrays and data statements
17. Flat feet
17. Quill Tip
17. QL repairs
18. Psion support
18. Slow Microdrive handling with Quill 2.00
19. QL Crossword review
20. Virtual windows
22. Sinclair repairs
23. A couple of tips for Abacus
24. The Which Computer Show
25. Review of Shrub Databank and Planner
27. Miracle Systems 512K RAM expansion
28. Games reviews - Master Blaster, Knight Flight and Flight Simulator
29. Data input routine
34. MEDICation
34. Review of Archiver
35. Stop press

INFORMATION ON THE GROUP

Membership of QUANTA (QL Users and Tinkerers Association) is by subscription to QUANTA, the group's newsletter, which is published monthly. Membership details are obtainable from the secretary. Membership of the group is open to anyone with an interest in the Sinclair QL microcomputer.

Members requiring assistance with problems related to the QL may write to or phone the secretary. An attempt will be made to put them in touch with a member who can help with the problem.

Workshops will be arranged from time to time in various parts of the country.

The group maintains a software library. Most of the programmes are free to members. A library list will be published from time to time. Programs are obtainable from the sub-librarians.

A list of members in any particular area is obtainable from the secretary.

Please send all contributions to the newsletter to the editor.

Chairman and
Newsletter Editor

Leon Heller,
65, Flanders Mansions,
Flanders Road,
London. W4 1NF
Tel: 01-994 7976

Secretary and
Newsletter Publisher

Brian Pain,
24, Oxford Street,
Milton Keynes MK11 1JU
Tel: (0908) 564271

Sinclair, QL, QDOS, QLUB and ZX Microdrive are trade marks of Sinclair Research Ltd.

Quill, Abacus, Archive and Easel are trade marks of Psion Ltd.

NOTES FOR CONTRIBUTORS TO QUANTA

Quill files on Microdrive or 40 track disk (single or double sided), are preferred. Cartridges and disks will be returned ASAP. Printed or hand written material is acceptable, of course.

Short programmes can be included, but lengthy programmes should be placed in the library, and a description sent in for the newsletter

.EDITORIAL

Some months ago we ordered Xpert Software's Nominal Ledger module, since when we have heard nothing, in spite of several phone calls to their answering machine. I am beginning to wonder whether their software actually exists. Has anyone else been able to get anything out of them?

QL World has purchased QL User from EMAP, and the next issue of their magazine will be entitled "QL World incorporating QL User". Paul Coster, the former editor of QL User will be joining the publishers of QL World, which will continue to be edited by Geof Wheelwright.

Syd Day tells me that the software librarians are getting very few requests. It would appear that whilst lots of members have got cartridges one and two, no-one seems interested in cartridges three to 28. Some of the material on the early cartridges was a bit simple-minded, I must admit, but the later ones have some really excellent programmes, most of which are FREE: So come on, make use of your library; you might like it.

We have just had a very successful meeting in Birmingham. Over 50 members attended, and the programme included sessions on DIY memory expansion, disk interfaces and DP's Supercharge compiler. Commodore demonstrated their new Amiga machine, for those members who were thinking of moving up-market.

We are thinking of organising a 68000 assembler course on two consecutive Sundays in Birmingham, in a few weeks time. We'll be sending details to members within easy traveling distance, but if anyone else would like to attend please give Brian a ring. The course will be quite relaxed and friendly, and will cover the 68000/68008 processor architecture, the instruction set, writing simple assembler programmes, and using QDOS. Members should bring their machines, and an editor/assembler package, as the course will have a large practical content.

A weekend workshop is being organised for the 15/16 March at the Wiltshire Hotel, Swindon. You should receive a provisional programme and booking form with this issue of Quanta. If you want accommodation advance booking is essential.

Leon Heller

MEMBERS' LETTERS

According to an interview given by Sinclair officials (Bill Jeffrey, Chief Executive, Charles Cotton, Director of Sales and Marketing, Duncan S. Lowrie, European Sales Manager) to the German magazine "Happy Computing" (Issue 1/86, January 1986, p 15) the present low price of the QL in the U.K. is only to be maintained for a limited time, in order to gain a substantial market share! After that the price will be upped again. Unfortunately, they didn't say when, nor by how much.

I feel I must write about some great after-sales service received from Micro-Peripherals. During one of my visits to England some months ago I bought their disk drive interface in one of London's Tottenham Court Road computer shops. (This was before the interface was adopted by Sinclair as their official badged version). Much to my dismay however, when I tried it here, it did not work properly in the Microdrive emulation mode, and then it did not work properly at all. This is always a problem when living in a country other than where you bought the goods. In despair, I just put the interface in a strong envelope and sent it to Micro-Peripherals, with a copy of the proof of purchase of the interface and a letter explaining the trouble (and discreetly mentioning that I was a QUANTA member). About eight weeks later (a not unreasonable delay, considering the distance) I received the interface back, in a beautifully packaged box with a new and updated ROM version, which now works without the slightest hiccup. I did not have to pay anything, not even the postal charges for sending me the repaired interface, nor any repair charge. I must say that I was delighted with this service which unfortunately is all too rare in the computer industry and one thing is for sure: If I ever want to buy any other hardware add-on and I have the choice between a Micro-Peripherals device and any other, I will chose the M-P one without any hesitation.

A small tip. When writing an SBASIC programme, it often happens that one is obliged to change the same line over and over again, i.e. one must 'EDIT xx'. In this case, one can put the following line in said programme: 1 EDIT xx,10. The programme runs normally, but once the programme stops, i.e. when an error is encountered, or when a STOP statement is executed, the line xx is automatically displayed in window 0, ready to be edited.

Wolfgang Lenerz,
146, Rue du Theatre,
75015 Paris.

This is the tale of a disillusioned computer user. Long days ago, around the time Noah was taking sailing lessons, I was tempted to spend £99.95 on the fabulous ZX80 with its massive 1k of memory. It didn't do anything useful but did that matter? Seemingly months later along came the Johnny-come-lately ZX81 at half the price which actually seemed to do something.

I was not to be tempted: Maybe something better would come along in as many months. Lo and behold, the Spectrum arrived. Was I to take the Spectrum leap? No, nothing short of a quantum leap would be adequate. Then a Quantum Leap appeared at £399.95, knocking the BBC into a cocked hat, especially with the Psion software. Fortunately, by the time I got around to making the decision to buy a QL everything had been fitted inside. The initial problems I had with the Microdrive cartridges were minimal, but version 1.00 of Quill especially, almost resulted in the whole shebang being hurled through the window.

Salvation ,was at hand. I have had no more problems with the cartridges and I received the full suite of version 2.00. What joy to produce real documents and actually to print them! Shame about Archive dropping the first record in the file and none of the programmes resetting after a "quit".

Then we had a magazine devoted to our machine (I must admit I like it) but I thought it would provide a Microdrive exchange. Five letters and £9.75 to Priory Court produced less than I got free from the Quanta library. More letters and hassle and I eventually got a subscription.

So I wanted to get to grips with Archive. Let's turn to IQLUG (it sounded like the bath emptying, QUANTA is much more pleasant) and bought Chas's Archive notes only to find them to be notes in the way that the Finance Bill is a note on public spending. Never mind, Leon (Issue 2.6) tells me that they aren't for beginners and I should have received a refund. Still waiting, Leon.

All this left me needing some light relief. Bravo, Medic Data Systems, for offering all those games for just £2.50. The advertising for their disk drive was so tempting I thought I would just get a taste of the software that went with it. Two cartridges and some weeks later I have one game that works (M-Treasure which is quite good fun, but I defy anyone to finish the game), one cartridge then returned to Medic and no sign of its return two months later. I won't go into CP Software (see Quanta 2.6 page 8)

I can think of no other industry, except perhaps Masochists' Mail Order, which treats its customers so offhandedly, sometimes *even* with contempt. So many promises with so little delivered,, I wonder why we all come back for more.

Yours in sadness

Ian McRobert
115, Park Road,
Peterborough PE1 2TR
(0733) 42127

/* Ian should have returned the notes to get his refund, Brian tells me. LFH */

A note to anyone who is thinking about buying a Bright Star from Modem House. After waiting 3 months for a refund cheque for a VTX5000 which never arrived (when the refund cheque got here, it bounced), I have been waiting for a Bright Star (for which I paid in advance) since October. The Managing Director told me last week that they have been having great difficulties with the delivery of parts and that these difficulties will take more than 4 weeks to sort out. It beats me why companies can't inform the customers via the press about such delays.

Bad news is better than no news at all and will prevent nightmares about such companies going bust and the loss of monies paid.

Dane Kurth,
Langgasse 51,
3292 Busswil,
Switzerland.

I am considering writing a BCPL interpreter, suitable for use with the Metacomco compiler, to assist with debugging programmes of any length. Would any QUANTA members be interested?

I have found a way of slowing down any game that is a bit too fast for comfort. Simply EXEC any suitable programme, such as the library disassembler, before running the game. If the game is run by using LBYTES and then CALL, the necessary EXEC must be inserted between the LBYTES and the CALL; otherwise the dummy programme may be EXECed at any time beforehand.

I'd better stop before I become a terrible bore, but thanks to all the regular contributors, I'm forever learning more from QUANTA:

David Palfrey,
Coxboro' Cottage,
Cookham,
Berks. SL6 9HR

For the first time in eighteen months I am able to see the whole of a letter as I write it using Quill. My Ferguson MC01 TV/R.G.B. Monitor has just been fitted with their part MA15 and the transformation to the display is miraculous. At last I have 85 columns (previously I had 75!). It costs £7.50 and can be ordered by your TV dealer from Ferguson.

Kevin Leah,
58, Mansfield Drive,
Merstham,
REDHILL,
Surrey RH1 3JN.
(07374) 4433

PROBLEM SECTION

I have a twin Medic system with 512 Kb. I wonder whether anyone could tell me how to use MSwitch??? I either have three Abacus (Abaci ?) and one Quill which locks up as soon I have "switched" more than once, or I get one Quill and one Abacus on the M-Switch screen which beeps at me rudely! I would be very grateful if someone could explain in step-by-step idiot language how to get ONE Abacus and ONE Quill selected. Postage and thanks will be paid with Swiss choccy if desired:

Could anyone tell my daughter how to use the parachutes in Caverns? There is no mention of this on the inlay card. (My daughter is better at Caverns than I am)

Dane Kurth,
Langgasse 51,
3292 Busswil,
Switzerland.

I notice in QUANTA 2,10 that Dennis Briggs was offering JS ROMs. I have an 'AH' QL and the serial number begins D09, can anybody tell me whether there are any other modifications that have to be made when installing the new ROM?

I have twice collected all 395 Diamonds in QL caverns, I then expected to be able to use the teleport on screen 0 but it remains barred. Can anyone tell me how you complete the game?

Kevin Leah,
58, Mansfield Drive,
Merstham,
REDHILL,
Surrey RH1 3JN.
(07374) 4433

/* Putting JS ROMs into AH machines is a bit tricky as a few wiring changes have to be made. If anyone has done this could they please write it up for Quanta. LFH */

Does anybody know how to get rid of the rubbish which accumulates at the end of a Quill document? (Try COPY xx_doc TO SCR to see what I mean). This seems to be all the errors made during typing 0 and which have been erased with the CTRL Cursor keys. As that rubbish takes up valuable storage space, on cartridge and in computer memory, I would like to know how to get rid of it. I tried to cut the document, by sending only part of it to an mdv file and later importing it, but it still contained the same rubbish. As I make typos rather frequently, my "overhead" is rather important. Have you got any idea?

Wolfgang Lenerz,
146, Rue du Theatre,
75015 Paris.

/* The "rubbish" at the end of a Quill document is, I believe, part of the document. It probably contains formatting information, which isn't stored with the text, unlike most word processors. LFH */

There had to be a first time! After using ABACUS, the original and Version 2.00 for the past 9 months, since I got my QL, I have never had any hold-ups or trouble, and I use the programme very frequently for a variety of personal business purposes investment, finances, invoices, etc. This morning I was happily adding the latest month's income and expenditure, with 7K still available, when mdv2 started and the screen went blank. It ran for nearly 10 minutes, before I became fed up and switched the machine off.

An attempt to load the particular file elicited the reply that it was not there: dir gave a no files answer. The back-up cartridge saved the day, but on attempting to copy the particular file to the cartridge that had been in mdv2_ using the ABACUS programme "bad or changed medium" was printed in green twice above the command position (not in white at the bottom of the screen).

I would be interested to know whether such problems, and others similar, which are comparatively frequent with QUILL, are a fault of the programmes or the machine. Even "Cartridge Doctor" would not have helped in this instance.

I would also like to learn whether there is any means by which I can prevent the annoying habit which my QL possesses of duplicating, and sometimes even triplicating characters when touch typing.

Gil Lamb,
Edifici Carles,
Esc. B, Atic B,
La Massana,
Principat d'ANDORRA.

/* If double characters only happen on a few of the keys, it is most likely to be a keyboard fault, and can probably be cured quite cheaply by replacing the membrane. If it occurs with all the keys, it must be something more fundamental within the machine, and will probably mean replacement of the I/O processor. LFH */

Reading in Quanta Vol 1, No 7, Page 26, with reference to QL locking out. I am having this happen with this model - D14. It seemed to happen after about an hour and a half, but since I obtained a Cleaner from Tony Firshman (pity I didn't know about the discount arrangement before I sent for it), now it seems to be somewhat more variable and I'm going to try to find if there's a sensible pattern - I've only had the cleaner for a few days. The machine just seems to seize up - the cursor disappears and I can find no way to persuade it to restart. The obvious solution is to save everything every half hour, and near the end of the time to do so more often.

This is the third QL I've had. My first was ordered in the first flush of publicity and it arrived early August 1984. First trouble was with the Psion programmes (blow this new spelling) they would not clone, two would not work at all. A new set was no better so the QL went back. Then, working through the User Guide I found about 19 programmes which would not work. As I'd not paid for it - probably a mistake on Barclaycard's part - I said I'd get one from a local supplier. Before I could do that, as I was a-lying in my bed I heard the news of the reduction in price. As I as to be in York a couple of days later I went in to Dixon's and asked about the microdrive extension - the bit on the right hand end, near the reset button, some folks don't know it is there! The assistant didn't know but promised an answer in 10 minutes. The extension unit for that end seems to be another of Sinclair's dreams as it has not yet been designed. There's some talk of being able to use a Spectrum extension, but I can't get much information about that, and there's some risk involved.

When I got the model home it was a D12, so I thought I'd done well. However, when I eventually looked at the keyboard - being a touch typist since 1932 I hadn't looked -- I found I had no key showing - and _ but had two with + and =, though using the keys in the place they ought to have been produced the correct character.

Another trip to York and the QL was exchanged for this one - a D14, with the difficulty that I have now got. So whether to return to York or to Sinclair direct I do not know. Any advice, please?

My thanks to John Tanner and Chas Dillon, both of whom have given me helpful advice and to you and others who must put so much time into producing Quanta. Disappointment in the member to whom wrote but didn't answer. Question! Anyone use Quill to keep details of the Register of Electors? Would be pleased to compare notes.

Arthur Nunn,
6, Westholme Road,
Masham,
Ripon,
North Yorkshire HG4 4EX

/* This problem with the QL "locking up" after being on for some time is almost certainly due to one of the chips overheating. It would be best to return the machine to Dixons, and get a replacement. I've had a similar problem with a machine we recently purchased as a backup - it went straight back to the supplier. LFH */

Could some bright machine code programmer write code to allow SuperBASIC extensions to be set up from languages like BCPL and C? This would also involve some way of passing the parameters of the extension to the high level language when the extension is called, for example in a BCPL vector.

David Palfrey,
Coxboro' Cottage,
Cookham,
Berks. SL6 9HR.

ARCHIVE BACKUP

As other members will have discovered, the ARCHIVE Backup procedure is not completely reliable, and the error "incomplete file transfer" sometimes appears on the screen. Of course the Backup procedure is far too slow to use with Microdrives and copying with SuperBASIC is much faster and more reliable. However with disks I find it more convenient to use Backup, keeping a backup copy of the file on the same disk, and later copying the whole disk onto a another one for safe storage in a second location. I use the following procedure within Archive:

```
proc backup
close
backup "data1_dbf" as "data_tmp"
kill "data2_dbf"
backup "data_tmp" as "data2_dbf"
kill "data_tmp"
open "data1_dbf"
first
Endproc
```

If the main file "data1_dbf" is corrupted during backup, then the existing backup file "data2_dbf" still remains untouched, and can be used to construct a new main file using SuperBASIC. Once the main file has been copied as a temporary file called "data_tmp", the original backup file can be replaced with a new copy, and the temporary file deleted. This process is still quite fast using disks, even though the backup process is carried out twice.

Alan Mason
3, Bransdale Road
Nottingham NG11 9JG

BOOK REVIEW : "INSIDE THE SINCLAIR QL" BY JEFF NAYLOR AND DIANE ROGERS, SUNSHINE BOOKS, £6.95

This book is aptly subtitled an introductory guide to the hardware: that's just what it is. It is more directed at the novice in computer hardware than the confirmed, soldering-iron swinging, electronic whiz kid, though even the latter may find many useful pieces of information in this book.

It is divided into two sections. The first one teaches you the "General Principles" of computer hardware design, i.e. starting from electrical current to logical gates, flip-flops, CRT displays etc... on to how a computer CPU works, RAM, ROM, address buses and decoding etc. All of this is covered in four chapters in which it is assumed that the reader does not know anything about the subject matter being treated. (Those that do, and who want information specific to the QL should jump right on to the second section). What I liked in this first section, was that the authors have, to

my mind, been able to find the right balance between the information given and that (necessarily) omitted. Thus they tell you everything you need to know - but not more, avoiding the cluttering of your mind with unnecessary details. Likewise, I very much liked the way the information was presented, with diagrams where there should be diagrams, and never in a condescending or paternalistic style. The authors just impart their knowledge on a certain subject whilst assuming that the reader who buys the book is bright enough to understand what is said therein.

In the second section, we move on to the QL. After a first chapter introducing the QL and all the chips it contains, the authors describe concisely the 68008 which is at the heart of the QL. Even though the book is not an assembly language primer, they then include a useful chapter presenting the assembly language instructions of the chip, in alphabetical order. This makes it clear (and the authors are at pains to stress) that the book is not intended to teach assembler, but the chapter is there if you want to look up a certain instruction. I found it useful. They even include a monitor programme in SBASIC.

Later on, in different chapters, they present input and output, the video display, the memory map etc, always looking at it at the hardware level, i.e., which chip is responsible for what, how does it interface with the 68008, how is it controlled, and so on... There is even a brief chapter on QDOS, traps and manager routines.

In summary, I found it a very useful book, well-written, clearly laid out and presented. If you are at all interested in the QL hardware, this is the book to buy. I was first lent a copy by a friend, but after reading it, I decided to buy it as I want to make sure to have it here when I need it.

Wolfgang Lenerz,
146, Rue du Theatre,
75015 Paris.

SUPERCHARGE REVIEW

Lots of members have expressed an interest in Supercharge, a compiler for SuperBASIC. I was sent a review copy some time ago, and although I have not been able to test the product as thoroughly as I would wish, my findings should indicate whether it is worth buying.

Supercharge, written by Quanta member Simon Goodwin, and marketed by Digital Precision, is a compiler that converts SuperBASIC programmes into machine language. SuperBASIC, like most BASICS, is an interpreted language. That is, each statement in a programme is first checked, to ensure that it is valid, and then executed, by the SuperBASIC interpreter, which is a large machine code programme contained in the QL ROMs. Compilers, on the other hand, take the entire programme, check that it is correct according to the "rules" of the language, and then translate the entire programme into another language, usually machine code, that may then be executed.

When the resultant machine code programme is executed neither the original programme (the source code) or the compiler, usually another machine code programme, have to be in memory. Compiled programmes run much faster than their interpreted equivalents because most of the error checking is only done once, when the programme is compiled. Because interpreters like SuperBASIC have access to the original programme, debugging and modification of programmes is very easy. On the other hand, programmes written in interpreted BASIC run rather slowly (because all the error checking is repeated each time the programme is run). Things are actually a lot more complicated than I have described, but you should get the general idea.

I tested version 1.11 of the package. Version 1.16 is currently being shipped, according to Simon Goodwin. The programme is supplied on Microdrive, with a 100 page manual. Whilst the programme is not copy-protected, a device called Lenslok must be used during compilation. The Lenslok system uses a sort of plastic prism in a plastic frame that is placed over a "scrambled" pair of characters displayed on the screen, which are unscrambled when viewed through the Lenslok device, and must then be typed in by the user. This is intended to prevent piracy. Compiled programmes are "stand-alone", and may be executed on any machine, although Digital Precision have imposed restrictions on the distribution of such programmes.

The manual is A4 sized, printed on yellow paper for some reason, and punched to fit the User Guide binder, or an ordinary ring binder. There is not really enough room in the User Guide and I feel that it should have been supplied bound in book form, or with a binder, considering the price. It is very comprehensive, clearly written, and suitable for most users, I would think.

The first thing to do is to back-up the cartridge, and this is made easy by the provision of a back-up utility, which configures the files for disk operation, if they are copied to disk. I used disks.

To compile a programme, it must first be loaded into memory. It's a waste of time compiling a programme that won't run under the interpreter, or has other bugs, so it should be checked thoroughly before compilation. A set of SuperBASIC extensions must first be loaded (this is best done by a boot programme) and then the Supercharge compiler itself loaded by the command "merge flp1_supercharge". The screen is cleared and a sort of scale is displayed, the size of which may be adjusted with the arrow keys to match the Lenslok device. No adjustment was needed with my monitor. When satisfied, <space> is pressed and the scrambled characters "OK" are displayed. The Lenslok device should be positioned so that "OK" is visible. <space> is then pressed again, and two scrambled characters are displayed, which should be visible through the Lenslok device.

The pair of characters must then be typed on the keyboard, within 10 seconds. If the characters typed are incorrect, one gets two more goes (different characters each time), and then the machine "locks up", and has to be reset. I must confess that I don't like this method of copy protection at all, and find that on average I have to have two attempts, resetting the machine in between. With practice, users will probably find the characters easier to identify. Using Supercharge with Microdrives would be extremely tedious, in my opinion, because of this protection method; it was bad enough with disks! Apparently a firm in Belgium has "hacked" Supercharge so as to remove the copy protection, and is selling pirated copies, but Lenslok should prevent casual copying.

Things are quite straightforward from now on. The user is asked for the name of the compiled programme, and if another programme of the same name exists, asked if it is to be deleted. One is then asked if a compilation listing is required, and if a report file is to be generated. The programme is then compiled (four passes) with the line numbers displayed whilst compilation is proceeding. Compilation was quite fast. If any errors are found, no machine code is generated, but compilation proceeds, so that all the errors are found in the one session. Warnings are generated if doubtful code is encountered, but compilation proceeds normally, and a machine-code programme is produced. This programme may then be "exec'd", as a normal machine code programme.

The compiler is somewhat "fussier" than the interpreter, but provided that programmes are written according to the QL User Guide, and follow the recommendations in the Supercharge documentation, users shouldn't have too many problems. I found what appeared to be a bug when trying Supercharge out on Clem Taylor's Othello programme from the library - it compiled without any problems but "bombed out at run-time at line 1570 with the error message "channel not open". I couldn't see anything wrong with the programme, which seems to run OK with the interpreter. Simon Goodwin is investigating the problem. Peter Dwyer told me he had problems using Supercharge with UDGs. Simon reckons that properly written DOG software (like Tony Tebby's) works fine with Supercharge.

The SuperBASIC extensions that are loaded before running the compiler fix some of the irritating bugs in SuperBASIC, such as the CALL bug, which causes problems when large programmes call machine language routines, and provide job control commands and allow the amount of free memory to be ascertained. The amount of data space assigned to a programme is fixed by the compiler, but may subsequently be changed using a utility called "dataspace_task".

The compiler generates "threaded code", which should be fast enough for most applications, and is quite compact. Faster, but bulkier, "in-line" code may be produced by enclosing the portion of the programme to be speeded up between "REMark +" and "REMark statements. Threaded code is used by most Forth systems,

and in fact, the code generation part of the compiler was written by Gerry Jackson, who has implemented Forth on the Dragon and QL.

In terms of performance, I would say that most programmes would be speeded up by a factor of between 10 and 30 times. A simple programme to calculate factorials took 22.6 seconds to calculate 300: with interpreted SuperBASIC, and 1.1 seconds when Supercharged. Other advantages are that compiled programmes load very much faster, cannot be modified or listed by the user, and may be multi-tasked.

Whilst compiled programmes may be used freely by the purchaser of Supercharge, compiled programmes may not be sold, unless a one-time licence is purchased, at a cost of £250. No royalties are payable. This seems like a reasonable arrangement, although very few other software houses now insist on charges for programmes produced with their compilers. I don't know how users stand who wish to give programmes away - I'll take this up with Digital Precision.

Summing up, I feel that Supercharge is an excellent product, except that I have reservations about the Lenslok method of copy protection. The price of £59.95 is not excessive, when compared with Lattice C and Pascal from Metacomco. Whilst I feel that C or Pascal are more suitable for serious applications programming, Supercharge is ideal for the ordinary user who wants his programmes to be more efficient, and who does not want to learn another language, as programmes may be developed and debugged very easily with the interpreter, and then compiled. This is not the case with C and Pascal, which are pure compilers (on the QL, at any rate).

My thanks to Freddy Vaccha of Digital Precision for supplying me with a copy of Supercharge, and Simon Goodwin for clarifying one or two matters. Digital Precision's address is:-

Digital Precision,
222, The Avenue,
London E4 9SE.
Tel: 01-527 5493

Leon Heller

ARRAYS AND DATA STATEMENTS

I will digress from the input routine I started in Vol. 2/10 to discuss arrays and the use of DATA statements, as I consider that virtually every programme can be more efficiently written using them. When I first had the rudiments of BASIC programming explained to me I was told that the programme used line numbers to determine the order in which the statements were executed.

It was not explained to me that DATA works independently, and it was perhaps a year before I recognised the way they are used. Arrays, or subscripted variables, describe a method of storing variables (string or numeric) in a sequence which is determined by the subscript or element number. Arrays may be single-dimensional, i.e. a simple list of items, or multi-dimensional, but we will only consider single-dimensional arrays for the moment. To benefit from the use of an array the items of data must have some relationship to each other, like the names of children in a school or the options to be given in the various menus of a programme, etc. Before we can assign the data items to the array elements we must declare that we require the storage space for the array to be allocated in the computer. This is done by including at a suitable point in the programme a DIM (dimension) statement. A string array may use any legal string variable as a descriptor such as a\$ or menu\$, say, and it is declared by the statement DIM menu\$(40,30), where the first number (40) is the length of the list minus one (it includes element zero) and the second number (30) the maximum number of characters we require for the longest item in the array (most other BASICS do not require the latter value to be declared). We could assign each element of the array by a direct assignment such as menu\$(3)="Load Saved Data" but that would be inefficient. Instead we use DATA statements in conjunction with the READ statement. To do this the data are entered in the programme at any point, but usually near the end, in the form :-

```
9010 DATA "COMPUTRONIC PROGRAM","Information","New Slide Sequence"
```

```
9020 DATA "Load Saved Data","Return to Main Display","Sort Slides into order"
```

The items may be separated on the same line with commas or as single items after the word DATA. They may be located within a PROCedure or separately; it will make no difference to the way they are read. The important point to note is that they must be in the order in which they are to be read into the array. I usually include a PROCedure named SETUP which contains all the DATA statements in blocks followed by a DIM statement and a READ statement for the preceding block. This helps to keep track of the number of items and to keep the READ statements synchronised. The form of the read statement for the above data would be:-

```
9030 z=5:DIM menu$(z,22):for i=0 to z:READ menu$(i)
```

The use of the variable z makes it more foolproof to change the size of the array if extra data are later added, as is usually the case when the programme is being developed. So the READ statement is equivalent to an assignment but it takes the data items in order. The operating system provides a pointer to the first data item when the programme is loaded, and this pointer moves to the next item when the first has been read. If the programme tries to read more items than have been entered it will generate an EOF (end of file) error message. SuperBASIC has the peculiarity that RUN does not perform a BASIC CLEAR and hence if the RUN statement is used a second time a RESTORE (or CLEAR) statement will be required before the first READ statement to reset the pointer to the first item. Now we have our data assigned to the array menu\$, how do we make efficient use of it?

I use it in two ways. Firstly, I have a simple PROCEDURE to write a heading to the screen. This uses the last selected menu function as the header name. I use the main programme name as the zero element which is displayed before any selection has been made. Secondly, I have a PROCEDURE to display a menu containing any number (up to 9) of consecutive elements of the array as options with, usually, the numbers 1 to n for selection. These procedures are mutually complementary.

```
10550 DEFine PROCEDURE HEADER
10560 MODE mo:CSIZE #5,2,0:at #5,1,(42-LEN(menu$(mu)))/2:PRINT menu$(mu)
10570 END DEFine HEADER
10580 :
10590 DEFine PROCEDURE MENU(y,x)
10600 mo=8:HEADER:po$=num$(1 to x)&"s":INK 7:PAPER 1
10610 FOR i=1 to x:AT 3+2*i,10:PRINT i:AT 3+2*i,14:PRINT menu$(i+y)
10620 AT 5+2*x,10:PRINT " enter choice or S top ":KEYINPUT po$:IF pos=x+1:QUIT
10630 mu=pos+y:INK 1:PAPER 7:HEADER
10640 END DEFine MENU
```

The MENU procedure would be called by a statement MENU 0,3 say. The header uses a previously defined window (#5). The menu assumes a full screen window i.e. WINDOW #1,512,256,0,0. The HEADER procedure clears the screen and writes the current menu\$ element defined by the variable mu centered on the screen. The MENU procedure displays the number of elements defined by the x parameter in the call and starts with the next element after the ' parameter. The variable po\$ used for our previous KEYINPUT procedure is sliced from the num\$ variable, previously assigned. To contain the to contain the numbers 1 to 9, with the addition of the 's' for the Stop option. It finally resets the variable mu to the number of the array element selected and calls the HEADER procedure to display the selected option.

Note that the number contained in the variable pos assigned in the KEYINPUT procedure happens to equal the number key selected, but only because the '1' occupied the first place in the po\$ and so on. With a slight variation one could easily assign a letter option in place of the numbers used in this case and the pos value would be just the same.

John Tanner,
43, Northumbria Drive,
Bristol BS9 4HL.
Tel:(0272) 623401

FLAT FEET

My QL has recently recovered from an attack of flat feet. When trying to fit the three plastic elevating legs recently, I discovered that the rubber feet had been completely flattened, and that the holes into which the feet should be inserted had closed up. The rubber appeared to be rather like a plastic pencil eraser in texture, and had been distorted by the weight of the machine and pressure on the keyboard.

Sinclair have now kindly provided me with free replacements, and the new feet appear to be made of much stronger rubber. If any other members have the same problem, I suggest they return one of the damaged feet to Sinclair and ask for a new set.

Alan Mason,
3, Bransdale Road,
Nottingham,
NG11 9JG

QUILL TIP

Probably the following is well-known to many QL users, but I have only just evolved it, and find it saves considerable hassle, when writing letters using QUILL. I have a couple of address cartridges (I have not yet got around to using ARCHIVE). With a new address I type that, save it, and print the envelope. I then load 'h', which is the file name of my home address, in one line across the top of the page to save space with a tilde at the beginning, and design needed for A4 paper. The tilde tells my Epson FX-80 via TRANSLATE in QUILL to print in Elite. I then type in the date, followed by merging in the address of my correspondent.

Gil Lamb,
Edifici Carles,
Esc. B, Atic B,
La Massana,
Principat d'ANDORRA.

QL REPAIRS

The following firms repair QLs considerably cheaper than Sinclair, and probably much quicker.

One Stop Micro Service Ltd.,
227, Tottenham Court Road,
London W1P 9AE.
01-631 0139

They might be able to fix your machine while-you-wait, if you take it to them. They charge according to how much work is required, and can do JS ROM upgrades for around £20.

Suredata,
45, Wychwood Avenue,
Canons Park,
Edgware,
Middx. HA8 8TQ.
01-951 0124

Suredata prefer machines to be sent to them. They charge a flat £20 plus parts, and tell me that the average cost so far has been about £25. They operate a 24-48 hour average turn-round time.

I can't guarantee the efficiency of these firms, of course, and would appreciate any feedback from members as to their expertise.

Leon Heller

PSION SUPPORT

Psion tell me that many users will find that their Psion support via QLAB has expired. Renewal now costs £28.75 per annum. QUANTA members will probably get better support by membership of the group, and shouldn't waste their money.

Version 2.3 upgrades are available from Sinclair, Camberley, at £7.50 per package. No new documentation is being issued.

Leon Heller

SLOW MICRODRIVE HANDLING WITH QUILL 2.00

In response to JR Catt's letter, I was describing Quill 2.00 when discussing slow Microdrive handling and he must have missed my earlier letter. I don't think there is any doubt, with my unexpanded machine at least (JM ROM, OS 1.03), that within Quill <BACKING-UP> is faster than <SAVING> by a factor of nearly three. An extra 128K does not speed up <SAVING> and slows down <BACKING-UP>, even when the whole document is within memory. This is disappointing because if one works with several copies of a document, as I do, saving for example a 3000 word document to three Microdrive cartridges would take about 10 minutes. Perhaps one needs 250K to get things really going.

There is a solution, and it has to do with ICE. ICE allows a RAM disk to be formatted and with a 128K memory expansion a 3500-4000 word document can be stored in RAM disk as well as being processed by Quill 2.00 entirely within memory. <SAVING> to RAM disk is quick and then <BACKING-UP> from RAM disk to Microdrive is quicker than directly <SAVING>, from within Quill by a factor of 50% and very much faster if one comes out of Quill. The latter makes sense if one wishes to save a large document to more than one cartridge.

ICE makes memory expansion more useful, and it is probably not worth getting anything less than 250K, although it would be interesting to have this confirmed. QUILL 2.00 obviously does not use extra memory to full advantage.

John Heckmatt,
17, Cleveland Court,
Kent Avenue,
Ealing,
London W13 8BJ.
01-997 3950

P.S. Any chance of encouraging someone to review Psion's DQRAW?

FOR SALE

Would any members be interested in purchasing a Miracle WS2000 modem, unused and in perfect condition? If interested, please ring 06285 22456.

David Palfrey,

QL CROSSWORD REVIEW

Supplied by:-

Quality Leader Software,
29, Layton Crescent,
Brampton,
Huntingdon,
Cambs PE18 8TS.

QL Crossword costs £12 inc. p&p

This aid to crossword-solving is cleverly written in SuperBASIC and supplied on two cartridges, one holding the working and help files, with a dictionary of 12,500 words, accessed on mdv2_ from within the program, on the other.

Its greatest strengths lie in flicking through the appropriate section of the dictionary (by word-length) and listing the words which will fit a partly-completed clue, and in anagram generation. You have to have some of a clue's letters before using the anagram facility (but it is possible to fool the program by entering random letters), and it lists on screen all possible permutations of the letters supplied - mostly rubbish. However, as it stands the list you see has the useful effect of suggesting the solution long before it makes its appearance: Selecting "Both anag/dic" very usefully compares with the dictionary and only displays real words.

Other facilities are comprehensive: crossword grids up to 15 X 15 can be defined, with symmetry in three possible planes; original compositions can thus be undertaken with input and amendment of clues and clue formats, or existing puzzles set up. In either case clue-numbering is automatic, though the numbers on the grid are rather hard to see in green on white. Crosswords can be saved, loaded and deleted; answers may be deleted leaving the letters of crossing clues in place. New words can be added to or removed from the dictionary, and hard copies taken.

Documentation, supplied on cartridge, is full and can be printed out. Commands are clearly prompted and responses to single keystrokes are rapid.-Some of the running instructions are rather confusing, particularly those for making backup copies and those indicating a change of cartridge, but the software is generally well up to spec; this reviewer, no crossword buff, found it effective and fun to use.

Dr. Basil Lee,
31, Fairdale Gardens,
London SW15 6JW
01-789 1976

VIRTUAL WINDOWS

I have a new contribution for the library. It is my system for handling virtual windows. The explanation is contained in the documentation file SCREEN1_DOC but please be sure to read and understand SCREEN_DOC before using SCREEN1_DOC. I have also supplied a game, as a demonstration of the techniques used, in order to make it easier to understand.

I hope the routines will be useful to members to create better QL demonstrations and games and to better understand computer graphics and some of the techniques used for various purposes. Brian Davies will decide whether the system should be supplied free of charge or will carry a royalty fee. I hope that members will find the system useful and informative and I would be particularly appreciative of feedback and comment from users as to improvements and/or new extensions in the same vein.

Before attempting to understand the use of virtual screens and windows, the user should already have a copy of, and an understanding of, the SCREEN COMPRESSION system by the same author.

This system of virtual screen handling, comprises the following files supplied on magnetic media and/or hard copy:

1. SCREEN1_ASM Assembler source code for the SuperBASIC extensions.
2. SCREEN1_CODE Assembler object code for the SuperBASIC extensions.
3. SCREEN1_DOC Documentation file for the above files, copyright notice and licence agreement.

This system contains the following extensions:-

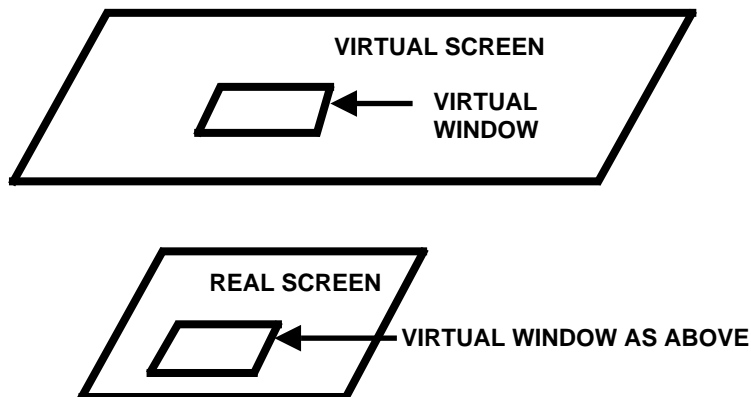
Procedures:

S_CLEAR x, y	Allocate & clear entire defined screen
S_SVON £n	Turn the screen save flag ON
S_SVOFF £n	Turn the screen save flag OFF
S_MOVE £n, x, y	Move to new screen position
S_SCROLL £n, d	Scroll up by d pixels
S_PAN £n, d	Pan left/right by d MOD 8 pixels (mode 4) or d MOD 4 pixels (mode 8).
S_RETURN	Return allocated memory
S_WINDOW £n, x1, y1, xb, yb, xv, yv	Establish window £n of size x1, y1 at position xb, yb with relation to the virtual screen and at position xv, yv with relation to the video screen.

Functions:

S_MEMORY	Returns address of allocated memory
S_SIZE	Returns number of bytes allocated

N.B. This system will currently support up to 16 virtual windows and each window must be an integral number of words wide. That is, each window must be rounded up modulo 16 on the x-axis, please also note that each window must be word-aligned on both the virtual screen as well as the actual video screen. This is to enable the system to operate as fast as possible.



The Virtual Screen is an area of memory allocated by the S_CLEAR procedure to be x pixels wide by y pixels high. A virtual window is created by a call to the S_WINDOW procedure to define the size of the virtual window as well as its co-ordinates on both the virtual and real screens. From this point on data movement is controlled by the save flags, S_SVON/S_SVOFF and the S_MOVE, S_PAN and S_SCROLL commands.

VIRTUAL SCREEN DEMONSTRATION PROGRAM

I have created a virtual screen demonstration program in the form of a game. The game, although perhaps not the most intellectually stimulating exercise ever created for a computer, will never-the-less give you a good idea of the power of the virtual window system. Perhaps it will stimulate the production of better and more varied games for the QL as well as opening doors for different kinds of graphic representations.

The game itself requires the use of the SCREEN COMPRESSION system on library cartridge 14. If you have one of the first versions of these extensions you will find that they will not operate correctly with the virtual window system. Please contact your local sub-librarian or the head librarian if this is the case and an upgraded copy can be provided. A slight modification was required on the original to allow screens to be larger than 32768 bytes. The new version was supplied and distributed during November 1985 and so should not effect members who have later versions.

To play the game itself, you simply load the SCREEN COMPRESSION system binary file (SCREEN), to a microdrive cartridge and RUN the program VIRTUAL BOOT from mdv1_. The main program will request the number of a screen file (0-3) and then proceed to load a picture from the microdrive cartridge. The picture will be displayed for a period of 1 second and will then be divided into 16 virtual windows. Each virtual window will then be moved to a new position on the virtual screen a random number of times. The final screen will look like a mixture of various segments of the original picture with parts of the original screen picture repeated a number of times.

The object of the game is to re-arrange the contents of the 16 virtual windows to re-create the original picture. This may be achieved by SCROLLing and PANning each virtual window by use of the cursor keys. The currently activated virtual window will be marked at the left side of the screen and at the bottom of the screen by a small red line. To change from one virtual window to another, simply press the Control (CTRL) key at the same time as one of the cursor keys, the red indicator lines will move and you will have access to a new virtual window.

I believe it is important to stress that this game is NOT the end product of the virtual window system, but MERELY a program to demonstrate the power available. I leave it up to people better qualified than I to create more imaginative uses for the virtual window system.

Ron Dwight,
Suvikuja 3B14
02120 Espoo 12
Finland.

SINCLAIR REPAIRS

My QL recently went from bad to worse and I decided to have it repaired, rather than do it myself as I have with other machines. On checking the adverts and asking around, there appeared to be at least four companies offering a variety of packages. However, I chose SINCLAIR. They offered a £40 fixed fee and 7-10 working days plus postage times. I delivered the machine by hand on Dec. 24th, paid by Access, and listed the following faults:-

1. Keys "W,f5" and others not working.
2. Microdrives unreliable.
3. One bit of video ram floating high by itself.
4. Cold boot RAM test fails when machine is warm.
5. "CALL" causes machine to hang with large program.

I got the machine back by post on Jan. 8th, one day after a postcard apologising for delay. It has a new keyboard membrane, 12 (out of 16) new RAM chips, the AH EPROM's have been changed to JM ROM's, and the Microdrives have only given one bad media message out of five copy cassette to disk tries which may be more reliable. They also included the full suite of version 2.3 Psion software - a bonus.

I am pleased with the work, and feel the rate is not unreasonable. It would be better if they had fitted the JS ROM, but you can't have everything.

David Stevenson,
Oman,
Muscat.

A COUPLE OF TIPS FOR ABACUS

ORDERING ON MULTIPLE KEYS;

Recently, I had an Abacus sheet which I wanted to order on three keys. Abacus only supports ORDERing on one key. If you order on one key and then another, the ordering on the original key is upset. It looked like I would have to export into Archive, which supports ORDERing on four keys. But this would be tedious. So I thought again. Here's a method.

Say that you wanted to order on two numeric keys (columns A and B), and the numbers involved were all less than 1000. Then create an extra column, column C, to ORDER on, which is a composite of the other two columns:

$$\text{col} = 1000 * A1 + B1$$

(This is for the case where column A is the primary key. If column B is the primary key, then simply use

$$\text{col} = A1 + 1000 * B1)$$

This idea can obviously be expanded for more keys, eg for ordering on columns B,C,A (in that order of precedence) you might create a composite key column:
$$\text{col} = 1000000 * B1 + 1000 * C1 + A1$$

In my particular case, the problem was a little more complex, in that one of my ordering keys was alphabetic. Abacus will ORDER alphabetically, but only on the single key. However, the above ideas can be extended, to take in alphabetical ordering, at least for a small number of letters. In my case, ordering on the first three letters would be sufficient.

To illustrate, consider ordering column A, containing text. Create another key, column B:

$$\text{col} = 128 * 128 * \text{CODE}((A1 (1))) + 128 * \text{CODE}((A1 (2))) + \text{CODE}(A1 (3))$$

This uses string slicing to take each of the first three letters in turn, converts them into ASCII code, and then multiplies by powers of 128 depending on their precedence (ie the first letter is multiplied by the highest number of powers). 128 is chosen because the ASCII codes will be less than this.

There is no point in using this method for ordering a single column, of course, but since you end up with a numeric ordering key, it can be combined into the method outlined above for ordering multiple keys. You end up with some large numbers, but let the computer worry about that:

MULTIPLE CHOICES:

The IF function can be used for making a binary choice; e.g.

```
IF(Sales>100,"good","bad")
```

Abacus does support nested IF functions, so you can use these for multiple choices; e.g.

```
IF(Sales>100,"good",IF(Sales>80,"OK","bad"))
```

Robert Matthews,
19, Trent View,
Marton,
Gainsborough,
Lincs. DN21 5AG.
(042 771) 308

THE WHICH COMPUTER SHOW

Since I was inundated with free tickets for the Which Computer Show, and according to Microscope, Sinclair had a stand there, I went along, hoping to see something new. Sinclair had apparently booked a stand, but had subsequently canceled it for some reason, which seems a bit silly of them, as their main competitor in the low-end business/home computer market, Amstrad, were there in strength. Sinclair's other competitor, Atari, weren't there either. There was very little to interest the QL user, it transpired.

PCML had a large stand, and had their disk drives for the OPD on show. They are using a Hitachi HD64180 (upwards compatible with the Z80) processor, with 256K RAM, controlling the disk drives, with a CP/M+ option, and MSDOS file read and write. I feel that they should have used an 8088 processor, and made it capable of actually running MSDOS. They couldn't demonstrate it to me, however, so it looks as if it isn't quite ready.

Prospero Software, who have just released their Fortran-77 for the QL, also had a stand. I should be getting a review copy shortly. They are now seriously thinking about developing a C compiler.

Although it didn't have anything to do with the QL, there was a very interesting "expert system" package on the Expertech stand. I've heard about this software, but it was the first time I'd had a chance to play with it. Written in Prolog, it enables almost anyone to set up their own expert-system, for medical diagnosis, electronic fault finding, Statutory Sick Pay, etc. Whilst nowhere near as powerful as the systems running on mainframes and LISP machines, it looks like some interesting applications can be developed. There is no reason why a similar package couldn't be put on the QL, of course, now that .5 Mb memory expansions are so inexpensive.

The Commodore Amiga received its UK launch at the show, and Metacomco, who adapted the Tripos operating system for it, were on the Commodore stand. I had a word with Dr. Tim King and Pam Clare of Metacomco, and heard that they will soon be releasing updated versions of their QL software.

The new Torch 68010-based Unix system was quite impressive, with a mouse-driven WIMP shell. I got one of the usual cryptic Unix error messages when I deliberately did something silly - I thought that those WIMP environments that are all the rage shouldn't do things like that.

Leon Heller

REVIEW OF SHRUB DATABANK & PLANNER

Shrub Databank and Planner is published by SUPERPLANT SOFTWARE at £19.95.

The program comprises two Microdrive cartridges: one contains the Archive run-time program and the other a data base file, containing in this instance details of 287 shrubs. Several other dbf's will be available on different botanical subjects. For those unfamiliar with "run-time", it is a modified Archive from Psion made available to software developers to provide a nucleus for them to build on their own procedures.

After loading, the main menu offers four choices: Selection, Display & Print, Re-set files and Quit. When Selecting, a further three choices are available which the program calls Enquiry, Habitat and Planner.

Enquiry: this allows you to search through the dbf for any characteristic (e.g. clay soil, evergreen etc.) and you then return to the main menu to enter Display & Print mode to view those shrubs selected.

Habitat: here you enter characteristics of the site you have in mind defined by 7 questions and these are helpfully explained by the screen prompts. If you don't know all the answers you can enter "no choice" and the program will search the dbf selecting those shrubs meeting your requirements. Once done, you move again to Display & Print to view or print your selections.

Planner: this facility allows you to set your own specific likes or habitat requirements one at a time, which operates by setting up a temporary file of those shrubs with the selected features. It is easily used as the screen displays a wide choice of feature adjectives from which to select. To further select you move back to the main menu before again accessing Planner. You can do this repeatedly. When finished, only those shrubs (if any) meeting all specified requirements will be left in memory.

There is approximately, 15K - 20K of available space on the dbf cartridge to allow one's own choice of additional plants to be stored. Alternatively, you can delete some of those supplied to create more space. The notes explain how this is done and there would be nothing to stop you creating your own complete data banks of different plants if you did not *wish* to buy further dbf's from Superplant.

Conclusion: the idea of this program is a good one and which Psion's Archive is well suited. However, I have a number of criticisms as I felt it fell short of the standard of which Archive is capable. Search times were slow, ranging from 75 seconds when using Planner (timing improved as the file became smaller with more selections) to over 3 minutes using Habitat.

Tighter programming could improve these times and take away some of the frustration. The working structure of the program, with its constant need to change screens to obtain the end result, was far too ponderous. Print choices are limited and it was tedious to have to repeatedly press the print command key to print-out the selected shrubs. Surely, the option to print "all" could have been given.

There is the basis of a good program here, but I do feel it needs sharpening-up in a number of areas before it could be recommended.

Alan Essex,
Rivington,
Rockfield Road,
Oxted,
Surrey RH8 OEL.
(088 33) 4796

MIRACLE SYSTEMS 512K RAM EXPANSION

There are a number of expansion RAM boards around, the cheapest per K I have seen is the PCML 512k for £99.50, but as I already had a CST disk system this would involve an expansion board as well. So I turned to the through connector type and of these Miracle's 512k for £125 appeared the best.

On phoning them I spoke to the most knowledgeable salesman of any company I have spoken to, I suspect he was also the board designer. He told me the only disk drive that wouldn't run with his board was the early version of the CST. The problem is that the buffers on the RAM card slow the signals too much for the disk interface, later versions of the disk board don't have this problem. The recommended solution was a double expander card which he would sell as a package at a discount rate. But as I was trying to avoid this he agreed it would work if I modified the board to have an unbuffered output. He was able to give me pin numbers and a description of the work required, and agreed to post the card to me the same day.

I have to report that the mod. works fine, but is not easy to do and I don't recommend it to people with poor eyesight.

The card plugs into the QL almost out of sight, however the disk card that now plugs into this extends 4 inches further beyond the case. To cover this gap a slightly nasty, shiny black plastic box is provided. The result is too floppy for lap use but is fine on a desk (or a bit of old plywood on my lap).

Is RAM expansion worth it? Well, my single drive disk copy program is much improved. The Psion packages have plenty of room, 490k free in Easel! BASIC runs all of 7 or 8% faster. But to be able to multi-task larger programs, flip screens in and out and generally not worry about memory size is the main object, and how well these work remain to be seen.

Miracle systems are on (0272) 603871.

David Stevenson,
Muscat,
Oman.

GAMES REVIEWS - MASTER BLASTER, KNIGHT FLIGHT AND FLIGHT SIMULATOR

MASTER BLASTER

This is the game that QL User gave nine out of ten. The ex assistant editor of the afore-mentioned magazine (Paolo Baccanello) was on Compugem's stand at the December Microfair and far be it for me to say that this is why it got such a good review. The graphics are the best I've ever seen on the QL (for a Galaxians clone), inventive, fast, varied and flicker free. BUT your average crippled snail moves with a better turn of speed and response time than the defending ship.

This is a game that was nearly the best I've ever seen for the QL but where it could have been a very exciting piece of software it only managed to be frustrating, but at £9.95 not bad value for money. I hope I've not put the programmer off; it has some of the most inventive graphic ideas out and if it had been released a year ago (when games were not possible on the QL) it would have made him both a fortune and a good name.

KNIGHT FLIGHT

This is one of the few games that even on the Spectrum would cost you (a well spent) ten pounds. The plot of Knight Flight (by Realtime Software, price £14.95) is that you're riding your trusty bird (in your armour hence the name Knight Flight) being attacked by and attacking the other bird riders. From this simple little plot comes the FIRST original quality game for the QL.

The graphics are excellent, the plot is well thought out and as both a game and a piece of software this one is a must for any games player. It's not only a challenge but (even at nearly fifteen pounds) good value for money. This is a quality of software that even game haters (is there such a beast) could admire, if only for programming technique. I hope Realtime Software has the success that they deserve with their first venture into the QL market and look forward to their next.

FLIGHT SIMULATOR

Those used to flight simulators on both the Spectrum and the Commodore will I think be a little disappointed by Microdeal's efforts here. As a piece of software its excellent, but rather than it being the game we'd all expected it's more of a training aid for would-be pilots. The controls are of a basic plane with nine inter-linked maps to fly over (with a choice of any to start with), good plane controls and wind control (just what I need), again with nine different settings. It has at least one novel bug which is that you can drive your plane like a car on land and even through the sea.

At £19.95 it's not cheap. But if you're looking for an interesting new piece of software to add to your collection I'd at least give this one a thought.

David Storton,
25, Trinity Rise,
London SW2 2QP.

DATA INPUT ROUTINE

Returning to the input routine started in Vol. 2/10, the next relevant procedure is used to obtain more than a single key stroke and echo to the screen the valid characters entered on the keyboard. It is called with parameters to define the maximum number of characters to be input and a string which contains all permitted characters.

The permitted characters need careful consideration. In addition to those which it is appropriate to allow within the returned input string they must include a terminator for the input, normally the ENTER key (chr\$(10)) and often one or more other terminators used to denote that a routine is completed by use of say the ESC key (chr\$(27)). A means of deleting unintended characters requires acceptance of a delete key, say, CTRL/cursor left (chr\$(194)). The terminators I assign to term\$ and I define num\$ as 'num\$="1234567890.- "&chr\$(194)&term\$1 and similarly for an alpha\$. The call statement may then be in the form GETCHRS 9,num\$. The strings may alternatively be combined in the call statement as GETCHRS 9,num\$sterm\$.

```
10110 DEFine PROCedure GETCHRS(n,ac$)
10120 Local i:INK 3:y$=":CURSEN:FOR i=1TO n
10130 AT y,x+o+i:KEYINPUT ac$:if z$ INSTR term$:CLS 4:EXIT i
10140 z=CODE(z$)=194:if z and i=1:i=0:NEXT i
10150 if z:i=i-2:y$=y$(1TO i):at y,x+o+i+1:CLS 4:NEXT i
10160 y$=y$+z$:PRINT z$:CLS 4
10170 END FOR i:INK 1:CURDIS
10180 END DEFine GETCHRS
```

The CURSEN and CURDIS calls enable and disable a cursor respectively (it will only blink in WINDOW #0). This function is available in the exten_code option on Library 1 and needs to be loaded and called in a boot program before the working program is loaded. The i loop operates to accept one character per iteration and hence if the maximum number of characters is entered the loop is EXITed without the ENTER key having been used. The variable y\$ contains the characters received (excluding any terminator). The loop is EXITed prematurely at 10130 if a terminator is used with less than the permitted number of characters. The logical statement assigning z at line 10140 permits a simple condition statement on the same and the following line. I will discuss logical statements of this sort in a later item. The purpose here was to separate the two situations in which the delete key may have been used from the normal valid key entry, in the first instance before any valid characters had been entered and the second when y\$ needed to be truncated, the last character erased and the cursor position moved back one position. The global variables y, x and o are used to define the screen line, the position on the line for printing the prompt (in a calling procedure) and the offset to the position of the initial input character respectively. The local variable i increments the position of the characters as they are entered. The normal function of incrementing y\$ and printing the received character occurs in line 10160. The CLS 4 is used to remove the unwanted character when the delete key has been pressed and to remove the default response printed by the calling procedure when applicable. This procedure also demonstrates the proper use of END FOR and NEXT, the latter only being applicable within the loop after a conditional statement. In this case as the counter has been decremented before the NEXT statement it could never EXIT the loop on lines 10140 or 10150.

The next requirement of our input routine is the means of providing the prompt against which the input is required and, as I mentioned initially, most serious programs require that the input data can be demonstrated in a hard copy form with the program output. I use the following procedure for this purpose.

```
10310 DEFine PROCedure OUTPUT(y,x,p2,i$)
10320 LOCAl le,p$,i:REPeat o_p
10330 p$=p2$ (p2) &FILLS (" ",0) :P$=P$ (1TO o) :AT y,x:PRINTp$;:le=len(i$)
10340 FLASH 1:PRINTi$:FLASH 0:GETCHARS 9,num$:AT y,x+o
10350 IF le AND y$=" ":PRINT:i$:EXIT o_p
10360 if y$<>" ":i$=y$:EXIT o_p
10370 END REPeat o_p:out$(p)=p$&i$:p=p+1
10380 END DEFine OUTPUT
```

You may have noticed that the GETCHRS procedure did not trap the null input, i.e. if the user only pressed the ENTER key the procedure was EXITed with y\$=" ". This would have been a simple matter to include in that procedure and if the above form of output is not required it would probably be the simplest course. However, as you will note, the above procedure uses the null return as acceptance at line 10350 of the default response passed as parameter i\$ to the procedure but only if i\$ itself is not a null. Otherwise line 10360 EXITs the procedure but only when a valid input has been received. You will also note that the procedure assumes an array in p2\$ has been filled with the required prompts and the element number is defined by the p2 parameter passed when it is called. The call for this procedure I use in the form OUTPUT 5,3,12,in\$(12) where the array in in\$ is DIMensioned in the SETUP procedure but is not filled. It acquires a valid string not by an explicit assignment but in consequence of the assignment if i\$=y\$ within the procedure, since the parameter of the call (in\$(12)) was not within brackets in the call statement the assignment of i\$ was equivalent to assignment of in\$(12)=y\$ and if the same call is made in the program later the induced string in in\$(12) will be printed as a flashing default response for the repeat input. The y and x parameters are, of course, the line and position at which the prompt will be displayed. The x value could be omitted by using a constant value of, say, three, which in mode 8 suits monitor or TV screens. You will appreciate that statements like IF le means the same as IF le<>0. In other words IF i\$ itself has a length of at least one character. The further array in OUT\$ must also be DIMensioned in the SETUP procedure to contain the output strings which will contain both the prompts used in the program and the responses given. The global variable p is incremented as each OUT\$ elemented is defined and must therefore be initialised with a value of zero before the first input is made.

All this may seem to contradict my "compact code" ethic. You may say that an INPUT with a message would have done. The benefit becomes evident when one looks at the working element of a typical program once we have these routines available. Suppose we wish to obtain responses to six prompts. The program would then read:-

```
1100 HEADER:y=4:FOR j=7 to 12:OUTPUT y,3,(j),in$(j):y=y+1
```

Limiting the useable keys to the numbers, decimal point and minus sign much reduces the possibility of an error when the number inputs now contained in the array in `in$` are subsequently used in some calculation. This would normally use the coercion feature of SuperBASIC such as `product=in$(6)*in$(2)` but the unfortunate omission of the VAL function from SuperBASIC makes this a fragile operation as the use of the minus sign other than once at the beginning of the number or the use of the decimal point twice in a number would create an error condition. Whilst a routine to trap such errors could be written, it would be far more lengthy than a simple condition used with the VAL keyword viz. `IF VAL(i$)=0 AND i$<>"0" THEN...`

If you will now indulge a few personal preferences...

Include in your SETUP procedure, or preferably your boot program, a `WINDOW #2,454,184,32,10` statement. This is the same as the IQLUG standard screen and allows a decent window for your listings. Do not write program lines which extend beyond this screen width. Not all printers can cope with listing programs with long lines and the only option is then to have an extra procedure in memory especially for the purpose of making a hard copy. It is also very tedious to edit very long lines and it is always quite likely that you will find it necessary to alter the order in which you first enter your DATA items to simplify their use in the program. A one line procedure such as:-

```
9000 DEFine PROCedure li:MODE 4:LIST:END DEFine
```

is most useful whilst writing a program and is called of course just by typing "li" and ENTER when you require to view a listing.

CTRL F5 pauses the listing as I hope you have discovered although, unbelievably, it does not rate a mention in the QL User Guide. Any key continues the listing.

When I first came to SuperBASIC I had no experience of procedures and I considered them as a direct substitute for subroutines. It quite quickly became evident that this is the wrong approach. The program should be written wholly as a series of procedures. I now have a standard first procedure named BOOT occupying one line:-

```
1000 DEFine PROCedure BOOT: RESTORE : SETUP : FROG : END DEFine
```

The programs are not LRUN from the boot program but are MRUN and the two program lines of the boot program which are merged with the main program make the call to the boot procedure. Even the RESTORE statement is unnecessary but is useful during development.

There are two distinct situations in the average program when we wish to display possible alternative options and obtain a response from the user to determine the next program step. The first is the formal menu display which was demonstrated by the MENU procedure, the other is when a function of the program requires the principal use of the screen be retained and the prompt **must** be displayed without losing significant screen space. The preferred procedure in the latter case is to use the bottom screen line only to display the prompt and in practice I define a window (#6) in which to print this form of prompt. Again the procedure uses arrays. The SETUP procedure requires a block of prompts and options thus:-

```
8170 DATA "C ontinue H ard copy","ch!"
8180 DATA "C ontinue R e-enter S top","crs"
```

followed by the necessary READ statement:-

```
8240 z=10:DIM prompt$(z,27),option$(z,4):FOR i=0 to z:READ prompt$(i) :
      READ option$(i)
```

so that we have related array elements in prompt\$ and option\$.

```
10380 DEFine PROCedure PROMPT(pr)
10390 CSIZE#6,2,0:CLS#6:PRINT#6,TO (42-LEN(prompt$(pr)))/2,prompt$(pr)
10400 KEYINPUT(option$(pr)) :CLS#6
10410 END DEFFine PROMPT
```

I omitted to mention previously the other important variable assigned in the KEYINPUT procedure, namely that of the numeric variable pos which contains the integer corresponding to the position of the character selected from the KEYINPUT parameter string. When the above procedure is called by PROMPT 3, say, and assuming that line 8190 contained the data for element number three of the prompt\$/option\$ arrays the value of pos could only contain the numbers one, two or three corresponding respectively to the selection of the "c", "r" or "s" keys in response to the prompt. The use of the numeric variable pos provides for the most simple means of branching in the program using perhaps a short form of the SElect statement thus:-

```
1210 PROMPT 3:SElect ON pos:=1:EXIT entry:=3:RETURN
1220 END REPeat entry
```

Assuming a "REPeat entry" structure as shown to allow the re-enter default to operate (i.e. when pos=2) it will exit the repeat loop on selection of "C" ontinue and further assuming that this section of the program is contained in a PROCedure it will leave the procedure by selection of the "S" top option.

This serves to show the clarity and brevity which is readily possible using SuperBASIC in conjunction with arrays and well considered procedures. Compare this with the all too common practice of placing a series of IF statements following any such input such as IF an\$='c' THEN ... etc. etc.

A further procedure which is frequently required is to provide a yes or no selection from a series of options. In the case now considered it is assumed that in any series of options if the earlier ones are rejected a final option is accepted by default. Again the option prompts are first read into an array. This time, rather unimaginatively I have used an array in pl\$. The selected option is also assigned to the out\$ array as used in the OUTPUT procedure.

```
10220 DEFine PROCEDURE YESNO (y,x,pl,b)
10230 REPEAT y_n
10240 p$=p1$(p1)&FILL$( " ,o) ,p$=p$(1 TO o):AT y,x: CLS 3:PRINT p$;"Y/N"
10250 IF p1=b:pos=1:ELSE :KEYINPUT "yn"
10260 IF pos=1:AT y,x+o:PRINT"Yes":EXIT y_n
10270 p1=p1+1
10280 END REPEAT y_n:out$(p)=p$:p=p+1:yn=p1
10290 END DEFine YESNO
```

The procedure is called with YESNO 5,3,16,4 for example. The first two parameters being the screen position for the prompt to appear the pl (16) parameter being the array element of the first option and the b (4) parameter being the number of successive prompts in the series which may validly be accepted, the last of which (array element 19) being the default choice. The selected option is returned by virtue of the assignment yn=p1 so that variable yn contains the array element number of the accepted prompt. As the value of y is not incremented and the CLS 3 statement clears the screen line only the chosen selection remains on the screen.

I will now return to the use of logical statements. These have a very wide range of uses and can eliminate the need for an IF statement in many cases.

The basic knowledge is that a logical statement which is true returns a value of one and a statement which is untrue returns a value of zero. Hence the assignment z=a=7 results in z having a value of one if the numeric variable a has the current value of seven but having a value of zero if its current value is not seven. we can alter the value which z acquires when the logical statement is true by z=(a=7)*5 so that z will now equal five or zero according to the truth of the statement within the bracket.

Apart from the first assignment illustrated, logical statements usually need to be within brackets to be unambiguous. Having grasped the fundamental significance of the logical statement it may be seen that they may be used in other than direct assignments. It is often useful to be able to select an array element according to the value of some variable, e.g.

```
FOR j=1 TO 6+(z=b):FRINT array$(j)
```

will print array elements 1 to 6 if z does not equal b and elements 1 to 7 if it does. Or we can include it in the subscript definition to shift the section of the part of the array to be printed:-

```
FOR j=1 TO 6:PRINT array$(j+(z=b)*6)
```

Thus it will print elements 1 to 6 if the logical statement is false or it will print elements 7 to 12 if it is true. The circumstances in which logical statements can be usefully used is limited only by ones imagination and remember that logical statements can equally relate to string variables as in the case of, say, $z=2+(a\$="q")$ which would assign a value of three if the logical statement is true and two if it were false.

I hope I have impressed how powerful is the combination of arrays with repeat structures and the use of simple procedures to avoid repetitive and wordy code. I appreciate that the listing of programs written using this technique may be obscure but it is wishful thinking that simple and direct code can also be efficient. It may be useful in a classroom situation that code is self documenting but efficiency demands a different style. I have recently been asked to consider marketing a program which contains 45K. In consequence it will not run on an unexpanded QL. It is evident from inspection of the code that it could easily be written in much less than a half of this space and hence its marketability would be much enhanced.

John Tanner,
43, Northumbria Drive,
Bristol BS9 4HL.
(0272) 623401

MEDICATION

Members with faulty MEDIC interfaces should contact Don Forbes on 01-688 5794. Don might have to make a charge for his services, but it should be quite reasonable.

REVIEW OF ARCHIVER

Archiver is marketed by Eidersoft at £16.95.

This is a series of specially written procedures to run with your own Archive program and which purports to provide: Invoicing, Stock Control, Appointment Diary

and Mail List/Mail Merge. It comes with a 48 page manual which is helpful in advising how you can change each program to suit your own needs. Also, there is nothing to stop you viewing how the procedures are built-up and perhaps Archiver could be helpful as a teaching aid for Psion's Archive. However, there the usefulness ends because the procedures proved far too cumbersome and their speed of operation could never be satisfactory even for the least serious user.

More worrying was the fact that several times programs crashed and after this I gave up taking Archiver seriously and instead examined, out of curiosity, how it was constructed. Members wanting programs for the above uses would be better buying a specific package. On the face of it you get four programs for the approximate price of a competitor's one but with Archiver it is very much a case of "you get what you pay for". I reckon Eidersoft will have lessened their image with those unfortunate to have already bought this. As already stated, perhaps it could be viewed as a teaching aid but even then it would be far too expensive. Not recommended.

Alan Essex,
Rivington,
Rockfield Road,
Oxted,
Surrey RH8 OEL.

STOP PRESS

In the January issue Ruben Zeida mentioned a problem with the Quill translate options. There is a mistake in the manual, and the option should read '27,"R",n' where 'n' is the ASCII code for the desired character.

Chris Stretch.

Can anyone help with a Quill printer driver for a Microline 83a (not the 82a mentioned in an earlier issue of Quanta).

Gordon Darling,
PO Box 3474,
Boroko,
Papua New Guinea.

It is worth repeating my advice to members to use a credit card for the purchase of goods (particularly hardware) with a value over £100, and insist on COD, or use a retail outlet, for goods with a lesser value. You can always give me or Brian a ring, and we'll tell you about problems with particular suppliers. Any members enquiring about Medic were warned to be very careful, right from the outset. It's a pity more of you didn't contact us. I know of a couple of other companies that are a bit "dodgy" but I can't print their names; we might get sued

Leon Heller

DATA MANAGEMENT

CLARK HOUSE THE VILLAGE FAXBY YORK YO5 8HJ
Telephone YORK (0304) 760847/760361

After a long wait virtually all of the GTE communications products have been added and we are pleased to announce some new packages.

TERMINAL EMULATIONS

Hard word (GTE/IBM) to 9600 baud, GTE/IBM to 2400 word depending upon term.

31 terminal packages are full implementations within the limitation of the GTE function key layout and Mode 2 screen. Time-line and message characters have been incorporated using colour and wrap changes. Full screen addressing and terminal times-schedule are included together with copying of messages for full use of all user-site features. A full manual for 24 line version is displayed in Mode 0. To take full use of this a monitor is recommended. Versions are available for 19 displays using only 24 lines.

For a small extra charge we will gladly configure your package to the word requirements of your host computer as you may prefer the extreme version using the PRINTER provided. Manuals normally requested are: 1 x 1 500 / 500 x 100 20/100 x 100 200 x 100 300 x 100 400 500 600 700 800 900 1000. Further enquiries details on request.

All terminal emulations: 15.95 + 1.50 (minimum of 1.00) + VAT.

PROTEL & SAS

These are PROTEL and PROTEL 542 nodes, so have now been brought through the GTE to release them to a wider user-base, with any additional hardware. The next arrival for this generation year by year to the DataTech team is a very well known name in electrical industry, and our software - which works very well to simulate the actual post-operative change the PROTEL is approved hardware, whenever it is available, the software is written to be used on performing well.

To give a better experience users to be able to a single unit, we have put together a total GTE communications package consisting of:

Hardware: 542 nodes plus
SOFTWARE: Full function VISITING / PARFORM software.
OPTIONAL: - 1 additional package with ASCII & XPODEM 91 + hardware.
SUPPORT: - Full technical package to qualify protocols and data flow details in the above.
CONSOLE: One dedicated GTE communication manual readability.
Serial Cable configured for Data.

Normally when prices went up on all hardware in excess of £2000 + VAT as a special for GTE we have now reduced the package for £2000 + VAT + 10.00. This is absolutely final and we are the only of our kind, we must say a minimum of 14 days delivery on the order. We will have units in stock for several requirements, but having now finally increased it to 14 days and not a return, and that you should be 14 days from date of order. We will be happy for you to stop the payment if you are not totally satisfied with our software, the order of the delivery terms.

With or price anyone that enquiries are sent to our 2400 baud 2400 baud DataTech.

RON D. SMITH, F.I.A.P.